SMT Encoding of Maximal Causual Model for Race Detection

1 SMT Encoding of Maximal Causal Model

$$\Phi_\tau = \Phi_{mhb} \wedge \Phi_{lock} \wedge \bigwedge_{e \in \tau} \Phi_e \tag{1}$$

where:

$$\begin{aligned}
\tau &= \text{trace of events} \\
\Phi_{mhb} &= \text{intra-thread program order and inter-thread synchronization constraints} \\
\Phi_{lock} &= \text{lock mutual exclusion constraint} \\
\Phi_e &= \text{feasibility constraint of event } e
\end{aligned}$$

$$\Phi_e = (O_e > M) \vee \Phi_e^{conc} \vee (\Phi_e^{abs} \wedge O_{next(e)} > M) \tag{2}$$

where:

$$\begin{aligned}
O_e &= \text{order variable of event } e \\
M &= \text{order variable representing infinity} \\
\Phi_e^{conc} &= \text{concrete feasitibility constraint of event } e \\
\Phi_e^{abs} &= \text{data-abstract feasitibility constraint of event } e \\
next(e) &= \text{event that immediately follows } e \text{ within the thread}
\end{aligned}$$

$$\Phi_e^{abs} = \bigwedge_{r \in dep(e)} \Phi_r^{conc} \tag{3}$$

where:

$$dep(e) = \text{all read events that precede event } e \text{ within the thread}$$

$$\Phi_e^{conc} = \begin{cases} \Phi_e^{abs} & \text{if } op(e) \neq read \\ \Phi_e^{abs} \wedge \Phi_e^{sc} & \text{if } op(e) = read. \end{cases} \tag{4}$$

where:

$$\Phi_e^{sc} = \text{read value observability constraint under sequential consistency memory model}$$

$$\begin{aligned} \Phi_r^{sc} = {} & \big(initVal(x) = v \ \wedge \bigwedge_{w \in W_{-}^{x}} O_w > O_r\big) \vee \\ & \bigvee_{w \in W_v^x} \big(\Phi_w^{conc} \wedge O_r > O_w \wedge \bigwedge_{\substack{w' \in W^x_{\bar{\ }} \\ w \neq w'}} (O_{w'} > O_r \vee O_{w'} < O_w)\big), \\ & \text{if } r = (read, x, v, tid) \end{aligned} \tag{5}$$

where:

$$\begin{aligned} initVal(x) &= \text{initial value stored in memory location } x \\ W_v^x &= \text{set of write events that write value } v \text{ to memory location } x \end{aligned}$$

Note: $initVal(x)$ is known upfront at the time when the formula is built so $initVal(x) = v$ does not appear in the implementation. And the problem stays in the "boolean formula over partial-orders" fragment.

2 Optimizations

2.1 Simplify Φ_e^{abs}

$$\Phi_e^{abs} = \bigwedge_{r \in dep(e)} \Phi_r^{conc} = \begin{cases} true & \text{if } dep(e) = \varnothing \\ \Phi_{lastRead(e)}^{conc} & otherwise. \end{cases} \tag{6}$$

where:

$$lastRead(e) = \text{last read event that precedes event } e \text{ within the thread}$$

2.2 Simplify $\bigwedge_{e \in \tau} \Phi_e$

$$\bigwedge_{e \in \tau} \Phi_e = \bigwedge_{\substack{e \in \tau, \\ op(e)=read}} \Phi_e \tag{7}$$

Proof sketch: If $lastRead(w) = \perp$, then $\Phi_w = true$. Otherwise, $\Phi_w = (O_w > M) \vee (O_w < M \wedge \Phi^{conc}_{lastRead(e)})$ and $\Phi_{lastRead(w)} \implies \Phi_w$. Therefore, Φ_w is always redundant.

3 SMT Encoding of Race Condition

$$\Phi_{cop(e_1,e_2)} = (O_{e_1} = O_{e_2} < M) \tag{8}$$

where:

$$cop(e_1, e_2) = \text{conflicting pair consisting of event } e_1 \text{ and } e_2$$